

CLAIMS

WHAT IS CLAIMED IS:

1. A method of efficiently decoding MPEG video to implement a reverse play or slow rewind function comprising:
 - receiving said MPEG video;
 - determining the type of said MPEG video received;
 - arranging a portion of said MPEG video received into one or more sequences of frames;
 - characterizing each of said one or more sequences of frames using one or more parameters;
 - decoding said one or more sequences of frames based on said one or more parameters; and
 - storing said frames into a number of video frame buffers, said number of video frame buffers implemented based on said type of said MPEG video received.
2. The method of Claim 1 wherein said determining, arranging, and characterizing is performed by a first subsystem.
3. The method of Claim 1 wherein each of said one or more sequences of frames comprises a feeding session.
4. The method of Claim 3 wherein said feeding session is stored into a first portion of memory.
5. The method of Claim 4 wherein said memory comprises a random access memory.
6. The method of Claim 1 further comprising displaying said frames from said video frame buffers onto a display.
7. The method of Claim 6 wherein said video frame buffers are implemented in a second portion of memory.

8. The method of Claim 7 wherein said memory comprises a random access memory (RAM).

9. The method of Claim 1 wherein said decoding is performed by a second subsystem.

10. The method of Claim 9 wherein said second subsystem comprises a second digital logic circuitry and a second software module.

11. The method of Claim 9 wherein said one or more parameters is used by said second subsystem to perform said decoding.

12. The method of Claim 1 wherein said type of video comprises high definition (HD) or standard definition (SD).

13. The method of Claim 1 wherein said MPEG video comprises MPEG I or MPEG II video.

14. The method of Claim 1 wherein said number of video frame buffers implemented is inversely proportional to frame resolution.

15. The method of Claim 1 wherein said number of video frame buffers is increased by the ratio $(1920 \times 1088) / (720 \times 480)$ when said type of video received corresponds to SD MPEG video.

16. The method of Claim 1 further comprising parsing said MPEG video into one or more segments prior to said arranging, each of said one or more segments comprising one or more frames, wherein only the first frame of each of said one or more segments is an I frame.

17. A method of efficiently decoding MPEG video bit streams to implement a reverse play or slow rewind function comprising:

determining the type of MPEG video received;

implementing a first number of video frame buffers based on said type of MPEG video received.

18. The method of Claim 17 further comprising generating one or more feeding sessions based on said first number of video frame buffers.

19. The method of Claim 18 wherein a second number of frames displayed in each of said one or more feeding sessions is limited to said first number of video frame buffers.

20. The method of Claim 19 wherein each of said one or more feeding sessions comprises a minimum number of I, P, and B frames required to adequately decode and display said second number of frames.

21. The method of Claim 17 wherein said implementing a number of video frame buffers maximizes the use of an available memory space.

22. The method of Claim 17 wherein said type of MPEG video received comprises HD or SD.

23. The method of Claim 18 wherein an algorithm is used to determine one or more parameters that define said one or more feeding sessions.

24. The method of Claim 23 wherein said one or more parameters comprises a first number of frames to be decoded, a second number of frames to be displayed, a third number of reference frames to be displayed, and a fourth number of B frames to be displayed.

25. A system of efficiently decoding SD MPEG video bit streams to implement a reverse play or slow rewind function comprising:

- a first subsystem capable of generating one or more feeding sessions; and
- a second subsystem capable of decoding said one or more feeding sessions received from said first subsystem.

26. The system of Claim 25 wherein said first subsystem comprises:

- a data feeder; and
- a first software module.

27. The system of Claim 26 wherein said data feeder comprises a first digital logic circuitry.

28. The system of Claim 25 wherein said second subsystem comprises:
A MPEG decoder; and
a second software module.
29. The system of Claim 28 wherein said MPEG decoder comprises a second digital logic circuitry.
30. The system of Claim 25 further comprising a memory used to store said one or more feeding sessions.
31. The system of Claim 25 further comprising a memory used to store a maximum number of decoded frames provided by said one or more feeding sessions.
32. The system of Claim 25 further comprising a memory, said memory implementing a maximum number of video frame buffers, said video frame buffers capable of storing decoded frames provided by said one or more feeding sessions.
33. The system of Claim 25 wherein said first and second subsystems are implemented using a system on chip (SOC), said SOC employing memory resources provided by a unified memory architecture (UMA).
34. A method of efficiently performing a reverse play of MPEG video comprising:
determining a first number of frames to be decoded from a feeding session;
determining a second number of frames to be displayed from said feeding session;
determining a third number of reference frames to be displayed from said feeding session; and
determining a fourth number of B frames to be displayed from said feeding session.
35. The method of Claim 34 further comprising determining a fifth number of video frame buffers to be used in a memory, said fifth number determined by the type of MPEG video provided by said feeding session.
36. The method of Claim 35 wherein said type of MPEG video comprises HD or SD.

37. The method of Claim 35 further comprising:
allocating said fourth number of B frames to said fifth number of video frame buffers;
allocating a first difference of said fifth number and said fourth number of video frame buffers to a second difference of said first number and said fourth number of reference frames;
performing a first decode;
discarding a third difference of said first number and said fifth number of decoded reference frames from said video frame buffers;
allocating said third difference of unallocated reference frames to said video frame buffers; and
performing a second decode of said video frame buffers.

38. The method of Claim 37 further comprising displaying frames from said video frame buffers onto a display.